Vision Transformers in 2022: An Update on Tiny ImageNet

Anonymous authors
Paper under double-blind review

Abstract

The recent advances in image transformers have shown impressive results training on large datasets like ImageNet-21k and then finetuning on smaller datasets such as ImageNet (and it's variants) and CIFAR-10/100 but have left out Tiny ImageNet in their experiments. This paper offers an update on vision transformers' performance on Tiny ImageNet. I include Vision Transformer (**ViT**) , Data Efficient Image Transformer (**DeiT**), Class Attention in Image Transformer (**CaiT**), and **Swin Transformers**. In addition, **Swin Transformers** beats the current state-of-the-art result with a validation accuracy of **91.34%**.

1 Introduction

The ViT paper (Dosovitskiy et al., 2020) showed that transformers can be applied to image classification tasks. However ViT was pretrained on the JFT-300M dataset (Sun et al., 2017), Google's internal dataset of 300 million images. Thus the issue of training efficiency and data availability was apparent. DeiT (Touvron et al., 2020) was a response to that and showed that a way to alleviate the data-hungry nature of transformers was with a rigorous training schedule and knowledge distillation. As such, it became possible to train a vision transformer using ImageNet-21k (Ridnik et al., 2021) and further finetuning on ImageNet-1k (Russakovsky et al., 2014). Subsequent image transformers, like CaiT (Touvron et al., 2021) and Swin (Liu et al., 2021b), closely follow the blueprint laid out by DeiT.

In addition to ImageNet-1k, these studies do transfer learning tests on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) but much of the focus is on improving ImageNet-1k accuracy.

Smaller datasets do have its purposes. It is much easier to overfit on a smaller dataset, therefore they can indicate how well a model generalizes. In addition, they offer fast empirical results and are a gateway into image classification for researchers who do not have access to the computing power needed to train on larger datasets. However every paper has failed to include Tiny ImageNet (Le & Yang, 2015), a subset of ImageNet-1k with 100,000 images and 200 classes.

A similar study has been done by **?** where they propose modifications to vision transformers to improve the accuracy training from scratch on smaller datasets like Tiny ImageNet. In reality, transfer learning is a much more common and stronger technique when it comes to accuracy. As such, this paper address that gap and will report the accuracy of **ViT**, **DeiT**, **CaiT**, and **Swin transformer** using a training regime similar to DeiT's.

2 Experimental setting

All vision transformers are taken from the timm library (Wightman, 2019). I trained each model using a Nvidia RTX 3070 (8GB memory) and an 8-core CPU. The size of the models were chosen based on being within the neighborhood of 10 to 60 minutes to train per epoch. For example, I could not afford to train a ViT-H/16 in a reasonable amount of time. As a refresher, I report the accuracies of the transformers on ImageNet-1k in **Table 1**.

Model	ImageNet-1k	CIFAR-100	CIFAR-10
ViT-L/16	87.08	**94.04**	99.38
DeiT-B/16⚗	85.43	91.40	99.20
CaiT-M/36	86.05	93.10	**99.40**
Swin-L/4	**87.15**	-	-

Table 1: Results of ViT, DeiT, CaiT, and Swin on ImageNet-1k, CIFAR-100, and CIFAR-10. All models are finetuned on 384x384 resolution. The numbers for ImageNet-1k are taken from the `timm` library and the rest are from the original papers. ⚗ signifies distillation. The authors of the Swin transformer did not report the accuracy on CIFAR-10 and CIFAR-100.

2.1 Data augmentation

The data augmentation techniques used mainly reflect that of DeiT's. I use Random Augment Cubuk et al. (2019) with 2 random ops and a magnitude of 9, Mixup (Zhang et al., 2017) and Cutmix (Yun et al., 2019) with a probability of 0.8 and 1.0 respectively, and Random Erasing (Zhong et al., 2017) with a 0.25 probability. I train using a random crop 384x384 resolution of the image and use a crop percent of 1.0 at testing time with Bicubic interpolation in both training and testing, as suggested in Touvron et al. (2020).

2.2 Regularization & Optimizer

For regularization, I employ label smoothing with an ϵ of 0.1 and a stochastic depth of 0.1

I train each model for 30 epochs, using 128 batch size. With the image resolution being 384x384, 8 gigabytes of video memory is not enough to load both the model and batch into GPU memory. As such, gradient accumulation was required to train with a 128 batch size.

The optimizer of choice is AdamW at an initial learning rate of 10^{-3} with cosine decay and weight decay of 0.05.

3 Results

Model	Tiny ImageNet	#params	FLOPs
ViT-L/16	86.31	304M	190.7B
CaiT-S/36	86.64	68M	48.0B
DeiT-B/16⚗	87.35	87M	55.5B
Swin-L/4	**91.34**	196M	103.9B

Table 2: Analysis of ViT, DeiT, CaiT, and Swin accuracy and training efficiency on Tiny ImageNet. I report the highest validation accuracy obtained during training. Throughput is measured with a batch size of 32. This is largest possible batch size that can fit on all models.

Swin The large Swin transformer achieves state-of-the-art accuracy of **91.34%**, beating the previous by 0.32%. Applying a window to multi-headed self-attention (MSA) and a shifted window to MSA proves to be effective. Swin continues to impress among the vision transformers.

ViT The ancestor of vision transformers, ViT, falls behind its advancements. The `timm` library reports that ViT outperforms DeiT by a significant margin on ImageNet-1k. Perhaps with adapted hyperparameters, ViT would see more benefits. On the other hand, Touvron et al. (2020) does report that ViT-L performs worse then a distilled DeiT-B so this result is not unexpected.

DeiT The base distilled DeiT achieves a respectable accuracy of **87.35%** while training the fastest by a large margin. The power of knowledge distillation is evident as it performs better then ViT-L and trains the fastest by a large margin.

CaiT CaiT has an accuracy of **86.64%**. Considering that CaiT-S/36 is based off a DeiT-S, the accuracy is expected. A CaiT-M/36 model would likely surpass the DeiT model but CaiT took the longest to train despite the parameter count and number of FLOPs.

3.1 Parameters and FLOPs are not created equally

CaiT shows that parameter count and FLOPs are not indicative of model efficiency. Despite CaiT-S/36 having the lowest parameter count and FLOPs, it reports the lowest throughput and trains the slowest, see **Table 3**. Instead, there is a trend with layer count and throughput. The size of the embedding is another thing to consider but the number of layers seems to be the main determining factor considering CaiT's small embedding size.

Model	**#layers**	**Embedding Size**	**#params**	**FLOPs**	**Throughput (images/sec)**
ViT-L/16	24	1024	304M	190.7B	31.5
CaiT-S/36	36	368	**68M**	**48.0B**	24.0
DeiT-B/16⚗	**12**	768	87M	55.5B	**83.1**
Swin-L/4	18	**192**	196M	103.9B	36.0

Table 3: Comparison of various model size metrics with throughput.

4 Tuning the training procedure

Swin's state-of-the-art performance with a standard experiment setup led me to believe further gains could be had with some further tuning. This section describes the experiments I did with the training procedure setup to achieve a final validation accuracy of: **91.35%**.

4.1 Hyperparameters

Optimizers For AdamW, I tried a combination of learning rates and weight decay in the range of [3.10^{-3}, 10^{-3}, 7.10^{-4}, 5.10^{-4}] and [0.2, 0.05, 0.01] respectively. I found the original setting of 10^{-3} learning rate and 0.05 weight decay to work the best.

Perturbed optimizers like SAM (Foret et al., 2020), ASAM (Kwon et al., 2021), and PUGD (Tseng et al., 2021) were also considered. Initial testing showed that SAM, ASAM, and PUGD increased the training time of an epoch by 85% while only having accuracies around 60-70% for the first 5 epochs. In comparison, AdamW has 89% accuracy after the first epoch. As a result, I decided not to train with these optimizers.

SGD was also tested with a learning rate of 10^{-2}, weight decay of 10^{-5}, and momentum of 0.9, with and without nesterov momentum.

Figure 1 shows that SGD converges faster then AdamW, likely due to the fact that I use a higher learning rate for SGD. On the other hand, AdamW is more variable then SGD perhaps because of the adaptive nature of AdamW. Either way, SGD falls short of AdamW in terms of accuracy but nesterov momentum performs better than vanilla momentum (91.21% vs 91.1%).

4.2 Ablation Study

This section describes the effects of removing various data augmentation and regularization techniques as well as trying different ones to determine what the optimal training setup is for Swin on Tiny ImageNet.

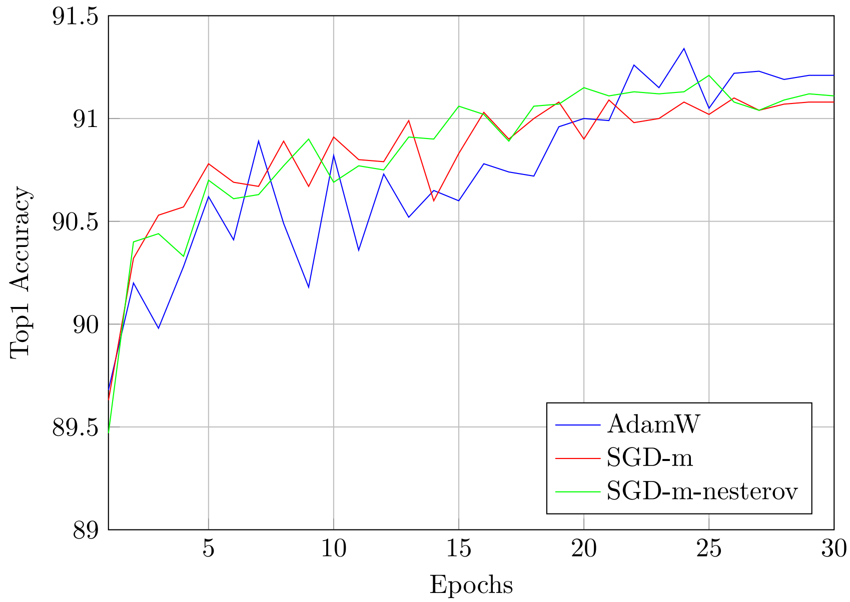

Figure 1: Convergence of AdamW vs SGD

Experimentation was done by simply removing one factor at a time and comparing their performance with the original setting. For new augmentations or regularizations, only one was added at a time onto the default configuration and compared that way. Techniques that did not prove to be useful were removed and any new techniques that improved accuracy were added.

I tried AutoAugment as it was a popular data augmentation technique before the introduction of RandAugment and an EMA (exponentially moving average) of the model weights.

Ablation	**Mixup**	**CutMix**	**Rand-Augment**	**Rand-Erasing**	**Stoc. Depth**	**AutoAugment**	**Accuracy**
Default	✓	✓	✓	✓	✓	✗	91.34
No Rand-Augment	✓	✓	✗	✓	✓	✗	91.35
No Rand-Erasing	✓	✓	✓	✗	✓	✗	91.28
No Mixup	✗	✓	✓	✓	✓	✗	91.11
No CutMix	✓	✗	✓	✓	✓	✗	91.04
No Stochastic Depth	✓	✓	✓	✓	✗	✗	-
Add AutoAugment	✓	✓	✓	✓	✓	✓	-

Table 4: Random crop is the only constant data augmentation technique. ”✓” indicates usage and ”✗” indicates ablation. This table is inspired from Touvron et al. (2020).

Without RandAugment, the model trains to a 91.35% accuracy which is within margin of error. However, the training patterns exhibited were attractive. The model was able to maintain its top accuracy for multiple epochs whereas the default setting peaks at 91.34% and plateaus at around 91.20%. For that reason, I decided to remove RandAugment.

5 Conclusion

This paper has shown that vision transformers transfer well onto Tiny ImageNet which makes sense as it is a subset of ImageNet-1k. Two standout architectures are DeiT and Swin. DeiT reports a respectable accuracy while training the fastest by a significant margin. And Swin achieves state-of-the-art accuracy, beating the previous by 0.34%. Future work could be done on even more vision transformers. SwinV2 (Liu et al., 2021a) improves on Swin and scales up Swin using self-supervised learning technique and a residual branch among other things. Another is MiniViT (Zhang et al., 2022), who uses a combination of weight sharing and weight distillation to drastically reduce the parameter count and increase accuracy.

References

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719, 2019. URL `http://arxiv.org/abs/1909.13719`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL `https://arxiv.org/abs/2010.11929`.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *CoRR*, abs/2010.01412, 2020. URL `https://arxiv.org/abs/2010.01412`.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. *CoRR*, abs/2102.11600, 2021. URL `https://arxiv.org/abs/2102.11600`.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer V2: scaling up capacity and resolution. *CoRR*, abs/2111.09883, 2021a. URL `https://arxiv.org/abs/2111.09883`.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021b. URL `https://arxiv.org/abs/2103.14030`.

Tal Ridnik, Emanuel Ben Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *CoRR*, abs/2104.10972, 2021. URL `https://arxiv.org/abs/2104.10972`.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. URL `http://arxiv.org/abs/1409.0575`.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *CoRR*, abs/1707.02968, 2017. URL `http://arxiv.org/abs/1707.02968`.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020. URL `https://arxiv.org/abs/2012.12877`.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *CoRR*, abs/2103.17239, 2021. URL https://arxiv.org/abs/2103.17239.

Ching-Hsun Tseng, Liu-Hsueh Cheng, Shin-Jye Lee, and Xiaojun Zeng. Update in unit gradient. *CoRR*, abs/2110.00199, 2021. URL https://arxiv.org/abs/2110.00199.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *CoRR*, abs/1905.04899, 2019. URL http://arxiv.org/abs/1905.04899.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017. URL http://arxiv.org/abs/1710.09412.

Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. 2022.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *CoRR*, abs/1708.04896, 2017. URL http://arxiv.org/abs/1708.04896.